CMPT 777 Programming Assignment 1

Sihui Wang 301474102

1. Formulation of Graph Coloring Problem:

Suppose that we use Boolean variables $p_{v,c}(1 \le v \le N, 1 \le c \le M)$ to denote whether the vertex v is colored with the color c:

$$\begin{cases} p_{v,c} \to \top & if\ vertex\ v\ is\ colored\ with\ color\ c \\ p_{v,c} \to \bot & if\ vertex\ v\ is\ not\ colored\ with\ color\ c \end{cases}$$

For graph coloring problem, we require that every vertex is colored, which means that $\forall 1 \le v \le N$, the formula $F_{1,v} \coloneqq \bigvee_{c=1}^{M} p_{v,c}$ must be true:

$$\forall 1 \le v \le N, F_{1,v}: \bigvee_{c=1}^{M} p_{v,c}, F_{1,v} = \top$$

In addition, we require that every vertex is colored with at most one color, which means that $\forall 1 \le v \le N, \forall 1 \le c_1 \ne c_2 \le M$, the formula $p_{v,c_1} \wedge p_{v,c_2}$ must be false:

$$\forall 1 \le v \le N, \forall 1 \le c_1 \ne c_2 \le M, p_{v,c_1} \wedge p_{v,c_2} = \bot$$

Or equivalently, the formula $F_{2,v,c_1,c_2} \coloneqq \neg p_{v,c_1} \vee \neg p_{v,c_2}$ must be true:

$$\forall 1 \le v \le N, \forall 1 \le c_1 \ne c_2 \le M, F_{2,v,c_1,c_2}: \neg p_{v,c_1} \vee \neg p_{v,c_2}, F_{2,v,c_1,c_2} = \top$$

Moreover, we require that no two connected vertices have the same color, which means that for $G = (V, E)$, for any pair of vertices $(v_1, v_2) \in E$, we have $\forall 1 \le v_1, v_2 \le N, (v_1, v_2) \in E, \forall 1 \le c \le M$, the formula $p_{v_1,c} \wedge p_{v_2,c}$ must be false:

$$\forall 1 \le v_1, v_2 \le N, (v_1, v_2) \in E, \forall 1 \le c \le M, p_{v_1,c} \wedge p_{v_2,c} = \bot$$

Or equivalently, the formula $F_{3,v_1,v_2,c} \coloneqq \neg p_{v_1,c} \vee \neg p_{v_2,c}$ must be true:

$$\forall 1 \le v_1, v_2 \le N, (v_1, v_2) \in E, \forall 1 \le c \le M, F_{3,v_1,v_2,c}: \neg p_{v_1,c} \vee \neg p_{v_2,c}, F_{3,v_1,v_2,c} = \top$$

So, given the definitions of the sub-formulas:

$$F_{1,v}: \bigvee_{c=1}^{M} p_{v,c} \qquad \forall 1 \le v \le N$$

$$F_{2,v,c_1,c_2}: \neg p_{v,c_1} \vee \neg p_{v,c_2} \qquad \forall 1 \le v \le N, \forall 1 \le c_1 \ne c_2 \le M$$

$$F_{3,v_1,v_2,c}: \neg p_{v_1,c} \vee \neg p_{v_2,c} \qquad \forall 1 \le v_1, v_2 \le N, (v_1, v_2) \in E, \forall 1 \le c \le M$$

$$F_1: \bigwedge_{v=1}^{N} F_{1,v}$$

$$F_2: \bigwedge_{v=1}^{N} \bigwedge_{1 \le c_1 < c_2 \le M} F_{2,v,c_1,c_2}$$

$$F_3: \bigwedge_{c=1}^{M} \bigwedge_{(v_1,v_2) \in E} F_{3,v_1,v_2,c}$$

the graph coloring problem can be formulated as checking the satisfiability of the formula:

$$F: F_1 \wedge F_2 \wedge F_3$$

i.e., we can find a solution to graph coloring problem if and only if we can find an interpretation I of $p_{v,c}(1 \le v \le N, 1 \le c \le M)$ so that the formula $F: F_1 \wedge F_2 \wedge F_3$ is true under the interpretation I.

Here we choose to represent the formula F in the CNF form, this is because Z3 solver implicitly assumes the conjunction form. Since we want to use Z3 solver to automatically check the satisfiability of the formulas for us, we need to formulate the problem in such a manner that the

formula is represented in the CNF form.

2. Program Design:

First, we load the input file describing the graph and the colors:

```
File file=new File("./src/cmpt/input.txt");
Scanner in=new Scanner(file);
```

Then, we read the first line to obtain N (number of vertices) and M (number of colors):

```
String data[]=in.nextLine().split(" ");
int num_vertex=Integer.parseInt(data[0]);
int num_color=Integer.parseInt(data[1]);
```

Then, we can initialize the solver and the variables. Here `p[i][j]` denotes the variable $p_{i,j}$, where i is the index for vertices, and j is the index for colors.

```
Context ctx=new Context();
Solver solver=ctx.mkSolver();
BoolExpr[][]p=new BoolExpr[num_vertex][num_color];
for (int i = 0; i < num_vertex; ++i) {
    for (int j = 0; j < num_color; ++j) {
        p[i][j] = ctx.mkBoolConst("p_" + i + "_" + j);
    }
}
```

Then, we construct F_1 and F_2:

```
for(int i=0;i<num_vertex;i++){
    solver.add(ctx.mkOr(p[i]));
    for(int j=0;j<num_color-1;j++){
        for(int k=j+1;k<num_color;k++){
            solver.add(ctx.mkOr(ctx.mkNot(p[i][j]),ctx.mkNot(p[i][k])));
        }
    }
}
```

In the outer loop, we use `ctx.mkOr(p[i])` to denote $F_{1,i}$: $\bigvee_{c=1}^{M} p_{i,c}$.

In the inner loop, `ctx.mkOr(ctx.mkNot(p[i][j]),ctx.mkNot(p[i][k]))` is the subformula $F_{2,i,j,k}$: $\neg p_{i,j} \vee \neg p_{i,k}$.

Using `solver.add`, we obtain the conjunction of the sub-formulas and get $F_1 \wedge F_2$.

Then, we read the edges of the graph and add the constraints for coloring of the edges:

```
while (in.hasNextLine()){
      String edge[]=in.nextLine().split(" ");
      int num_1=Integer.parseInt(edge[0]);
      int num_2=Integer.parseInt(edge[1]);
      for(int i=0;i<num_color;i++){
          solver.add(ctx.mkOr(ctx.mkNot(p[num_1-1][i]),ctx.mkNot(p[num_2-1][i])));
      }
}
```

Here `num_1` and `num_2` are two vertices which form an edge. In `input.txt`, they are indexed from 1 to N. However, in the variable `p[][]`, they are indexed from 0 to $N-1$. So, we need to use `p[num-1][i]` to represent the variable $p_{num,i}$.

Here, `ctx.mkOr(ctx.mkNot(p[num_1-1][i]),ctx.mkNot(p[num_2-1][i]))` is the subformula $F_{3,num\_1,num\_2,i}$. By using `solver.add`, we obtain the conjunction of all constraints and get $F_1 \wedge F_2 \wedge F_3$.

Then, we use the solver to check the satisfiability of the problem.

```
Status status = solver.check();
```

We create a file `output.txt` to keep record of the results.

```
File out=new File("./src/cmpt/output.txt");
FileWriter writer=new FileWriter(out);
```

Then, we check the solver's status. If the status shows that the constraints are not satisfiable, then we simply output `UNSAT` in `output.txt`.

If the status shows that the constraints are satisfiable, then we need to go through the interpretation which satisfy all constraints. We need to find all variables $p_{v,c}$ with the assignments $p_{v,c} \rightarrow \top$. We output all $p_{v,c}$ with the assignments $p_{v,c} \rightarrow \top$ in `output.txt`, which is a feasible solution to the graph coloring problem. Please note that there is a difference in indexing between $p_{v,c}$ we previously defined and `p[i][j]` we created in the program. This indexing issue has been taken care of in the following code.

```
if(status.toString().equals("SATISFIABLE")){
    Model model = solver.getModel();
    for(int i=0;i<num_vertex;i++){
       for(int j=0;j<num_color;j++){
            if(model.getConstInterp(p[i][j]).toString().equals("true")){
                writer.write(Integer.toString(i+1));
                writer.write(" ");
                writer.write(Integer.toString(j+1));
                writer.write("\n");
            }
        }
    }
}
else if(status.toString().equals("UNSATISFIABLE")){
    writer.write("UNSAT");
    writer.write("\n");
}
```